

PassWindow: A New Solution to Providing Second Factor Authentication

*Simon Nettle, Sean O'Neil, VEST Corporation
Peter Lock, Cascade Blaze Pty Ltd*

Abstract: PassWindow is a new methodology for providing second-factor authentication in the online environment. It involves two segmented matrices – one static and one variable – which, when superimposed, reveal a set of digits that is used as a one-time password. This method provides robust security that compares favourably against all existing solutions according to a generally accepted set of criteria providing a sufficient increase in security over static passwords and PIN codes at a very low cost. The method can be utilized to provide transaction authentication information to the user making it one of very few presently available authentication solutions for protection against the latest security threats.

Introduction

This document introduces and reviews PassWindow, a new entrant into the field of secure online authentication methods. It claims to be a rather user-friendly solution that significantly increases security of authentication in the online environment at a very low cost, which makes it extremely cost effective and attractive. In this white paper, we will explain what PassWindow is, the problem it was designed to address, how it works, and how it compares to the existing authentication solutions.

In addition, we will look at the PassWindow security features, implementation, and why organizations may want to deploy PassWindow as their second-factor authentication methodology of choice from a business perspective.

What is PassWindow?

PassWindow is an internationally patented method of providing secure and robust authentication online. It involves no electronics, is relatively easy for customers to use, and at the time of writing is by far the cheapest of the available solutions. It would be most commonly implemented as the second factor in the authentication process, and used sparingly so as not to fatigue the customer by insisting on an extra step to authenticate.

Potential uses for PassWindow

PassWindow aims to protect online services requiring extra security afforded by two-factor authentication, such as:

- High security user logons
- Monetary transactions to new accounts
- Account creation and modification
- Proof of software ownership
- Online services vulnerable to fraud, such as credit card transactions

It can also be used to:

- Verify the authenticity of official documents distributed electronically, or even simple customer mail-outs
- Obviate the need for a password, so all the user needs is a username and their key pattern-embedded card

Indeed, PassWindow appears to be cost-effective for businesses to deploy across their customer bases and is likely to significantly mitigate the risk of online card-based fraud, thereby saving a great deal of trouble and monetary loss.

How does PassWindow work?

Implementation

In a nutshell, PassWindow involves a segmented matrix key pattern printed on a transparent region of a credit card sized plastic card (the key pattern) that the user superimposes with a similar pattern displayed on the screen (the challenge pattern) to reveal a series of digits (the solution). The user submits this solution, which is relayed to an authentication server that verifies that the user has the card in their possession.

The system itself consists of a PassWindow Authentication Server (PWAS) that issues valid challenge patterns and reports whether the solution entered by the user is correct.

Multiple One Time Passwords (OTPs)

While the key pattern printed on the card is static, the challenge pattern is varied each time the user is required to authenticate, creating a multitude of unique OTPs.

The human visual system can easily recognize the emergent digits among the noise in the superimposition. The PassWindow system also allows the amount of noise to be varied on an individual basis for users who have difficulty recognizing the digits.

The critical security benefit of this method of authentication lies in the fact that the components of the solution are never communicated over the same medium, thereby preventing a hacker from obtaining usable authentication information from a user.

The problem

At present, the Internet has emerged as the primary communication method in our modern lives. It will no doubt also become the predominant means by which purchases and other financial transactions are carried out. The rise of this technology has created a concomitant demand for personal authentication methodologies that securely and verifiably confirm the identity of the entity conducting the transaction in question. As authentication methodologies have become more sophisticated, so have the attacks.

Security

Even security consultants often fall prey to a common misconception that security is something absolute, expecting it to be either present or not. In fact, security is measured by dividing the difference between the cost of attacks and the benefit to the attacker by the cost of defending against them. Thus, the expensive albeit more secure methods such as cryptographic PKI devices with their own secure communication channels, screens and keyboards score so dramatically low on the security scale, while the world still predominantly relies on the cheapest and seemingly the least secure PIN codes and passwords. The total cost and complexity of deploying such devices often outweighs the benefit from their ultrahigh security.

Common attack methods

- **Compromised online databases** – collected information stored in merchant databases is stolen
- **Man in the middle / phishing** – a third party intercepts and impersonates the client and server to the respective other to record and/or alter their communications
- **Social engineering attacks** – customers are deceived into revealing their private details to a hacker

- **Man in the browser** – malware is installed on the victim’s computer to report network activity, keystrokes, and screen capture data to the attacker allowing interception during fund transfers in which funds can be unwittingly diverted by altering the information displayed in the user’s browser
- **Brute-force cracking of user passwords** – the server is polled with every possible password combination
- **Simple theft** – authentication details written down or on a card can be physically taken and copied
- **Shoulder surfing** – an attacker can surreptitiously watch the user enter their transaction details

The associated perception that the Internet is an unsafe environment on which to conduct sales and other financial transactions mitigates the utility of what is otherwise one of the greatest boons to merchants all over the world.

Current solutions have proven unsatisfactory due to their high cost or difficulty of use: high shipping costs due to the heavy weight, high hidden server licensing costs, batteries die unpredictably quickly, screens fog up or crack, electronic components rust in the heat and humidity common to many heavily-populated tropical countries, clocks go out of sync with the server, devices get washed inside the user’s pockets, their PINs get entered too many times by their children, etc., etc., etc. Meanwhile, even the most secure of the devices get hacked.

Thus, many organizations have turned to unsafe simpler authentication methods for the majority of their user bases, which has led to the proliferation of identity theft and fraudulent transactions in the online environment.

Two factor authentication

The notable rise in demand for robust authentication strategies has been met by the introduction of two factor authentication (TFA), which means employing two of the three commonly available methods of verifying an individual’s identity: requesting something they know along with something they have and/or something they are.

- **Something they know** – usually a username and/or password/PIN code
- **Something they have** – a physical item carried by the user
- **Something they are** – biometric data about the user in question

Generally, it is the first two of these that are employed in TFA strategies: a username and password (first factor) combined with a physical token (second factor). The effective combination of these two factors is sufficient to prevent almost all attempts at false identification.

However, finding a method of implementing the second factor that satisfies both the user’s and the vendor’s needs has hitherto proven elusive. The solutions to this problem that are currently being marketed all still present the prevalent problems of cost and/or difficulty of use.

Evaluation criteria

In evaluating the available TFA solutions, we used the following criteria:

1. Cost effectiveness
2. Ease of use for customers
3. Administrative burden (IT, administrative, logistic)
4. Portability
5. Longevity, durability, and reliability
6. Security in the online environment

This document evaluates PassWindow according to the above criteria and compares it to other competing technologies, *inter alia*, through its capacity to resist the common methods of attack outlined previously.

Existing solutions

Present solutions to the problem of online TFA generally provide a physical or quasi-physical token that is unique, difficult to obtain or forge, and can be reliably verified by the authentication server. Most produce a One Time Password (OTP) based on a shared secret between the user and the organization. The secret is either the token or is contained within the physical token.

These tokens take a few forms:

- Hardware tokens
- Software tokens
- Physical tokens and other methods

Hardware tokens

Hardware tokens are physical objects that utilize dedicated on-board electronics to generate OTPs. The dongles / key fobs presently employed by a number of banks are the most common examples of hardware tokens.

These tokens broadly employ two modes of operation: event-based and time-based password generation.

- **Event-based tokens** generate a new OTP according to a mathematical algorithm each time a certain event takes place, such as pressing a button or entering a PIN into the device. The details of the algorithm form the shared secret. Some event-based tokens use a challenge/response method whereby a user must physically enter a PIN or challenge code presented by the authentication server to produce an OTP.
- **Time-based tokens** issue a new token at specific time intervals – usually 18 or 30 seconds – and both the token and authentication server are time-synchronized such that each can intuit the other's state. Excessive variance in synchronization indicates that the system's security may have been compromised.

Hardware tokens are further divided into connected and disconnected kinds:

- **Connected tokens** require a connection to the client computer to transmit the authentication information. This is achieved either by physically connecting the token to a USB or similar port, or via a logical connection negotiated over a wireless communication technology.
- **Disconnected tokens** require the user to enter the token's information manually into their computer.

Software tokens

Software tokens employ similar methods to hardware tokens to achieve secure authentication, but they are stored on a general-purpose electronic device, such as a desktop or laptop computer, PDA, or mobile phone.

The advantage of software tokens over their hardware cousins is mostly that the per-unit cost of software tokens is significantly lower as the electronic platform on which they reside is usually something the customer already owns, such as their mobile phone.

Unfortunately, this benefit is offset by problems such as the lack of portability if the token is stored on a desktop PC, an increased vulnerability to malware in that it might be stored on an infected device, and compatibility issues – for example, due to the lack of a common specification among mobile phone manufacturers.

Physical tokens and other methods

Physical tokens

Other authentication technologies that do not rely on computational logic also exist. For example:

- **Number grids:** A printed grid of one-time passwords referenced by x and y coordinates
- **Transaction authentication number lists:** Similar to number grid, this is simply a list of “unguessable” OTPs

While these are effective and relatively cheap methods of online authentication, they lack durability and portability, are complicated to use and fail to convey an adequate sense of security to the customer.

SMS-based methods

A popular authentication method is to use an SMS message to deliver an OTP to the customer’s registered mobile phone number. Because the OTP is communicated over a completely separate channel, it is immune to being interfered with by malware installed on the user’s PC. However, as the mobile network is not secure, this is an example of security through obscurity. In addition, its reliability is limited by signal coverage and the necessity for the customer to possess a functional mobile phone.

Furthermore, as mobile phones grow in complexity, they become increasingly susceptible to malware infection themselves. This possibility is only likely to increase as this trend continues. The news of large active mobile phone based bot-nets consisting of tens of thousands of hacked mobile phones has already reached the press.

Biometric solutions

Biometric authentication involves verifying something about the customers themselves as the criterion for successful authentication. This might involve scanning a fingerprint, a retinal scan, or measuring some other exotic property of the human body unique to an individual.

Unfortunately, once the biometric data has been copied by a hacker, it is forever compromised and cannot be replaced. The fingerprints, for instance, can be compared to a permanent 3-digit PIN code, which must be the same for all the different applications, which one leaves on everything one touches, which can never be replaced but which can be used against its owner in court as evidence sufficient to convict one of murder. It is also not unreasonable to expect that in the future, DNA-based identity theft will become increasingly likely if such biometric identification becomes widespread.

Biometrics also lack one of the most important features required of an authentication device, the delegation of rights. In simple terms, the user of a fingerprint-protected car cannot let someone else drive it (while the keys are left for the thieves right on the door handle and all over the steering wheel). Forced to use such devices, people will quickly find ways to circumvent such terribly inconvenient security by faking their biometric signatures just so they could delegate their authentication rights to those they trust.

A percentage of the population cannot use biometric devices at all. In addition, recording the customer’s biometric data requires them to present themselves in person for the initial measurements, which is logistically difficult and expensive. The office and the staff required to perform these measurements is a significant additional hidden cost of the biometric authentication devices, none of which their manufacturers and distributors like to mention.

Chip Authentication Programme / Dynamic Passcode Authentication (CAP/DPA)

An authentication standard that has emerged recently, particularly in Western Europe and the UK, involves using the deployed ‘Chip and PIN’ method of second factor authentication. This is known as CAP/DPA (Chip Authentication Programme / Dynamic Passcode Authentication) with respect to the names applied by MasterCard and Visa, respectively.

The method uses a smartcard embedded in the user’s banking card in combination with a card reader to generate challenge-response authentication information that the user enters when conducting a transaction online.

While this method provides a reasonable defence against fraudulent transactions, its implementation renders it susceptible to a range of attacks, which were highlighted in a white paper produced by Cambridge University titled *Optimised to Fail: Card Readers for Online Banking* (<http://www.cl.cam.ac.uk/~sjm217/papers/fc09optimised.pdf>). The authors mention the following vulnerabilities:

- **Exposure to malware:** The card reader was intended to be a trustworthy external device to verify transactions away from any malware that might have infected the user's PC. However, the size of the reader makes it inconvenient to carry and has consequently generated demand for a software implementation. Since the reader contains no secret, it has been successfully reverse-engineered. It is expected that this demand will be met by software vendors leading to malware-infected PCs having unfettered access to the smartcards and their PINs.
- **Man-in-the-middle attacks:** Because the CAP reader can be easily tampered with, it cannot be regarded as a trustworthy display. A criminal can set up a tampered Chip & PIN terminal, which displays one transaction, but is actually relaying the smartcard communications to a counterfeit card being used for a much higher value transaction.
- **Supply chain infiltration:** Because CAP readers contain no secret information, and because their inherent physical bulk encourages customers to use the same card reader for all their online banking relationships, a market for generic CAP readers has emerged and with it so has the possibility of supply chain infiltration. For example, a cleverly-designed generic CAP reader could copy the card's chip details, record the user's PIN, record the magnetic strip, or even prompt the user to enter username and password information. All of this information could be relayed to the attacker when the reader is sent for repair after deliberately malfunctioning, through a mobile phone surreptitiously embedded into the device, or even online if the reader requests to be connected to the user's PC.

Supply-chain infiltration presents limitless opportunities to attackers, and is a crucial weakness of the CAP/DPA authentication method. The ease with which this could be done unfortunately negates the benefit of having a physically separate device on which to conduct authentication procedures.

The physical size of the readers presents its own problems. Users are reluctant to carry them on their person and their distribution across an entire user base incurs a significant delivery cost to the issuing organization for postage and handling.

The high cost of the provided security and the high potential benefit to the attacker have put such a high pressure on the CAP/DPA authentication method that its security is collapsing under its own weight.

Evaluation: PassWindow

In this section, we will evaluate PassWindow as a second-factor authentication method according to the criteria mentioned earlier.

Cost-effectiveness

For organizations choosing a TFA strategy, cost is often the primary concern. Indeed, the present high cost of providing the second factor has led to its being deployed reluctantly and only in relatively high-risk situations.

The overall cost comprises a number of factors. For example, organizations must consider the cost of:

- Producing or purchasing the physical item that is being distributed to customers
- Distributing the technology to the customer base
- The server and/or client software licensing fees associated with the right to use the technology
- Integrating the product into the organization's existing authentication infrastructure
- Providing ongoing administrative support
- Educating customers and staff in the use of the new technology

PassWindow is likely to be the cheapest of the available solutions with regard to all of the above factors. Beyond the licensing fees paid to purchase the right to use the PassWindow technology and its initial

integration into the existing authentication infrastructure, it introduces little to no additional expenditure with regard to production, distribution, or maintenance.

Ease of use for customers

Ease of use is often sacrificed in favour of greater security as organizations struggle to reduce the incidence of online identity fraud; for example, by insisting on the use of long, complex passwords that contain no known words and a variety of symbols.

While well intentioned, this is a critical mistake. Customers habitually resist complication; to introduce it encourages risky, security-degrading behaviours such as using the same password for multiple accounts or recording them in easy-to-find places (e.g., on the identity card in question or under the mouse pad).

This phenomenon has become so widespread that it has even earned a formal term: password fatigue. Therefore, any potential TFA solution should be as easy as possible for customers to use so as to not exacerbate their already-tested patience with online authentication systems.

PassWindow is a hitherto unseen methodology, and as such will likely require explanation to customers and the training of staff. However, the authentication process itself is simple and involves a real-world action – aligning a physical card with a pattern displayed on the screen – something that may be inherently more understandable and inviting to non-technically-minded customers.

Administrative burden

The administrative burden of the proposed authentication methodology is a key concern as it directly and significantly influences the cost of implementing the technology. The administrative burden is multi-factorial and traverses multiple departments within an organization.

The administrative burden organizations face can be broken down into the following general areas:

- Distributing the second factor to customers
- Providing helpdesk facilities to deal with lost, malfunctioning, or stolen tokens
- Maintenance, troubleshooting, and replacement of the distributed second factor
- Establishing and maintaining the required server-end IT infrastructure
- Training employees and customers in the use of the technology

PassWindow's administrative burden is likely to be quite low. From an IT standpoint, implementing PassWindow involves the installation of a PassWindow authentication server and the addition of a few lines of code to the authentication gateway to request a challenge pattern from the authentication server and to request verification that it is correct.

The necessary real-world security to ensure the safe transportation of user key pattern data to card manufacturers is likely to already be in place in most large institutions.

Organizations switching from existing solutions to PassWindow are likely to experience a reduction in administrative burden as they are liberated from having to maintain electronics in the field.

Portability

Acceptable portability is one of the key deficiencies of the existing authentication methods. If the token is bulky or otherwise difficult to carry, customers may lose it, store it in insecure locations, or invite theft by leaving it in the open.

PassWindow is the most portable of the available tokens. Almost everybody is accustomed to carrying standard, credit-card sized cards in their wallet, purse, and so forth. In addition to being easy to carry, being able to stow their PassWindow key pattern away from public view provides added security to the customer.

Longevity, durability, and reliability

Longevity: This refers to how long the device can operate in the field without requiring repair or replacement. Many of the available second-factor authentication devices are limited in their longevity by battery life, durability issues, or have redundancy engineered into their designs, which can significantly influence the cost of the solution.

Durability: Being physically robust reduces the cost and administrative burden of repair or replacement. Therefore, susceptibility to extremes of temperature and humidity as well as resilience to shock, submersion, and general physical mistreatment must be considered.

Reliability: This refers to how frequently problems will be encountered in the use the technology, both by customers and by potentially malfunctioning systems at both the client and server ends of the transaction.

The reliability of the solution is influenced by:

- **Malfunction** – possible in the case of tokens that utilize electronics or other digital logic
- **Factors complicating or limiting its use** – e.g., will signal coverage be an issue, as in the case of mobile phone-based tokens?
- **Reliance on associated technology** – must embedded systems or software-based solutions function correctly for the solution to work?

Due to the nature of PassWindow, it is unlikely to be the weak link in any malfunction, and it can be used anywhere that a challenge pattern can be displayed.

The only foreseeable reliability issue is customers misreading the key/challenge solution or being unable to differentiate the emergent digits from the obfuscatonal noise. However, the likelihood of this can be reduced by employing authentication management software that examines whether a customer is experiencing difficulty and then altering the challenge patterns to afford increased legibility for elderly and/or visually impaired customers.

With regard to these criteria, PassWindow can be considered to equal that of the customer's existing card. When a card expires, or is lost or stolen, a new PassWindow key pattern must also be issued. However, this involves no additional cost and little to no extra administrative effort.

Security in the online environment

Providing a secure and user-friendly authentication procedure to their customers is the primary goal of organizations implementing a TFA system. While attacks are generally made against single-factor authentication systems due to their comparative ease, TFA systems have begun to fall against new and ingenious hacking methods.

While it is regarded as impossible to provide an absolutely 100% reliable guarantee of robustness to attack, any authentication solution should render broad-scale attack impossible and dissuade the focused, motivated attacker.

In the next section, we will discuss how PassWindow can be used to protect against the latest and most destructive hacking techniques.

Methods of attack

PassWindow provides strong authentication in the online environment. As the 'something you have' element of the authentication strategy, the details of the customer's key pattern are never accessible to the host computer, and therefore cannot be stolen by electronic means. PassWindow provides a well-rounded set of defences against hackers in comparison to other methodologies.

Compromised online databases

Merchants who, despite best practice guidelines, retain Card Verification Value (CVV) or related information in their databases completely expose their users to credit card fraud should their user databases fall into the wrong hands.

Because the user's PassWindow key pattern data is never transmitted or stored electronically, and remains unknown and unrecorded by everything apart from the issuing organization's PassWindow authentication server, this crucial piece of information will always be lacking from user databases, rendering the information they contain useless to attackers.

Man in the middle / phishing

Man In The Middle (MITM) attacks involve a third party intercepting communications between a client and server, impersonating each to the respective other and intercepting, recording, and/or altering communications between them. Phishing is a kind of MITM attack that usually involves a fake login screen for well-known online services that reports login details to the attacking third party before seamlessly forwarding the user to their desired destination, unaware that their authentication details have been compromised to be used maliciously at a later date.

Because it is impossible for a third party to create a valid PassWindow challenge pattern without prior knowledge of the key pattern, users are immediately alerted to a spurious request by a garbled or absent challenge pattern. In this way, PassWindow provides authentication in both directions – from the client to the server and server to client (as in to the person, not to a workstation or device).

Social engineering attacks

Social engineering involves the customer being convinced to reveal their private details, and in the case of hardware tokens, their OTPs.

A PassWindow key pattern is not easily communicated verbally or by typing. It is very unlikely that an attacker would attempt to extract the pattern from the customer in this way, as they would simply be unable to explain the details of the pattern. With sufficient education, customers could be taught to never reveal the pattern to anyone and to keep their cards away from prying eyes.

Man in the browser / hacker infiltration

When an attacker receives reports from the malware installed in the victim's computer and detects that the victim is accessing their financial institution's website, the software alters the form data in the browser such that a different amount of funds are transferred to a different account – usually a 'mule' account. The owner of the mule account then transfers this money to the attacker.

Verification information about the transaction taking place can be encoded into the PassWindow challenge pattern. This can assure the user, for example, that the funds are being transferred to the correct account.

Simple theft

The only way for a PassWindow key pattern to be revealed and duplicated is by directly copying the card in one's immediate possession. This possibility is reduced by the introduction of a tint that can be printed over the pattern, hindering attempts at photography and photocopying.

However, because PassWindow should be used as the second factor in the authentication strategy, mere knowledge of the key pattern is insufficient for fraudulent authentication without also knowing the victim's username and password.

Shoulder surfing

While probably the most mundane of the 'hacking methods', PassWindow is secure against 'shoulder surfing' – surreptitiously watching the user enter their transaction details. Because the key/challenge solution is a one-time password, the shoulder surfer cannot benefit from knowing it.

Again, a tint printed over the key pattern on the card renders the pattern itself invisible to anyone but the user.

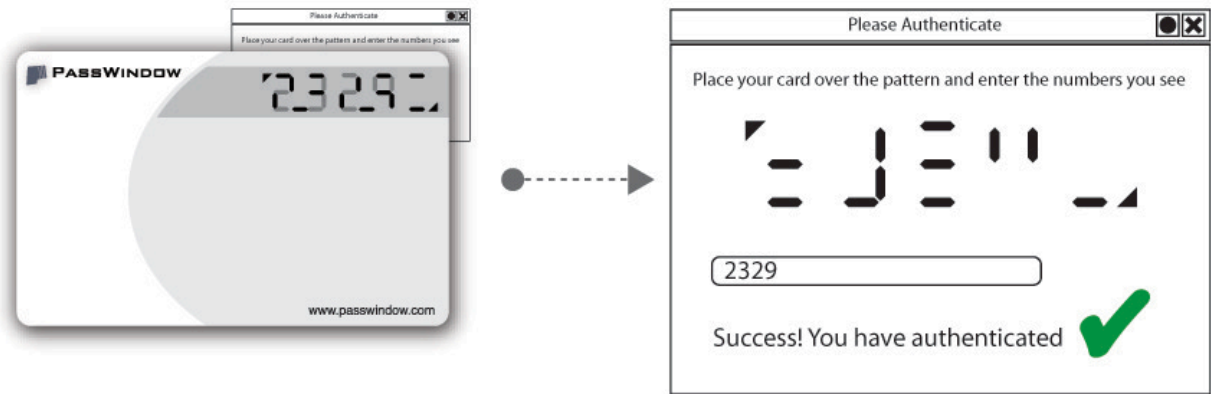
PassWindow – a closer look

How PassWindow works

PassWindow is a unique challenge/response authentication paradigm that involves a sequence of numbers displayed as a segmented matrix divided into two components that can only be revealed by superimposing the two corresponding patterns.

Each PassWindow-embedded card includes a transparent window with one half of the solution data, the key pattern, printed in the window using standard card printing technology. An authentication server co-located with the issuing organization's existing IT infrastructure possesses an array of challenge patterns and their corresponding solutions for each key pattern.

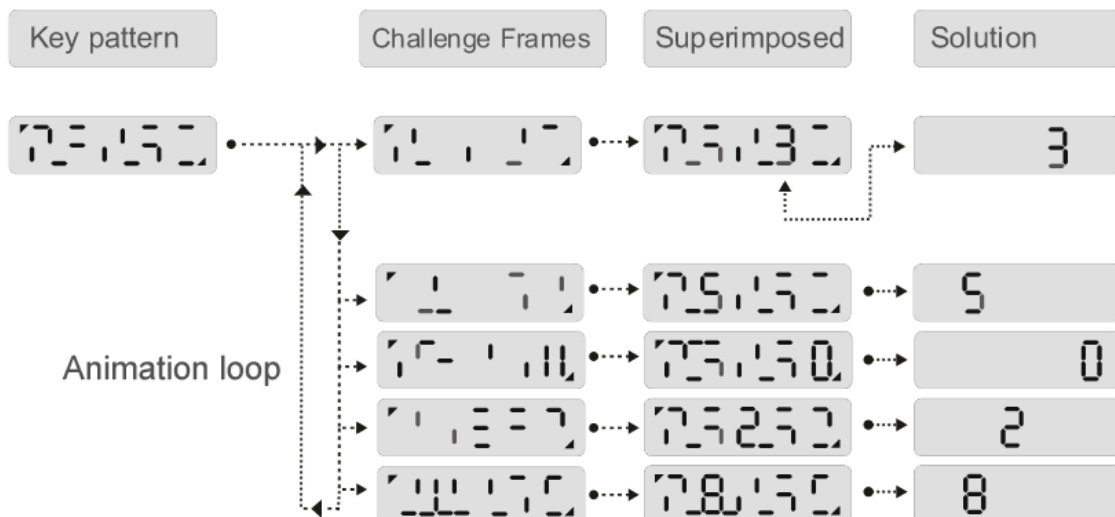
During the authentication process, the user is requested to superimpose their key pattern with the challenge pattern presented on the screen and then to submit the emergent digits:



These digits can be displayed either as a static string, where all digits are revealed in the one string, or as an animated sequence of single-digit challenges. The static string is used in print-based implementations, whereas the more secure animated method is recommended for electronic and online applications.

While the static pattern method presents all digits of the password in a single pattern, the animated method presents each digit of the password alone in its own individual challenge pattern. These single-digit challenge patterns are presented to the user as frames in an animation loop. The user then enters the digits in the order that they appear, beginning at any point in the looped series. This provides around two orders of magnitude greater security on account of the difficulty of analysis that the method introduces.

The animated method:



Once the user enters the unique one-time password, the authentication server checks whether the solution given was correct; and if so, reports that the user has been successfully authenticated.

How PassWindow is implemented

In the most basic sense, implementing PassWindow as the second factor in an authentication methodology involves:

- The physical installation of one or more PassWindow servers, depending on authentication load and necessary contingency measures
- Relatively simple programmatic additions to the organization's authentication web gateway
- Arrangements with card manufacturers to print the unique PassWindow key patterns on the cards

The user experience

PassWindow is most commonly implemented as the second factor in the authentication process, and is often used sparingly so as not to fatigue the customer by insisting on their making the extra step to authenticate. For example, a financial institution might elect to require only the first factor (a username and password) to log on, but require the second factor (PassWindow) when the customer wants to make a monetary or other sensitive transaction.

The user is shown the challenge pattern, prompted to superimpose the key pattern on their card with the challenge pattern displayed on the screen, and to enter the emergent digits.

The PassWindow Authentication Server then checks to see if the solution is correct, and if so, the authentication gateway grants access.

Recognition of emergent numbers

The numbers that appear when the key pattern and challenge pattern are superimposed will sometimes not be as discrete as, for example, the digits on an LCD watch.

This is to improve the security of PassWindow such that it provides the most robust and difficult-to-crack authentication. Because of the characteristics of human visual processing, it is easy to identify the numbers that appear among the obfuscational noise necessary to thwart hacking attempts in the same way that it is easy for humans to read CAPTCHA challenge-response tests.

Security features

Combating hacker infiltration

Hackers are potentially able, using sophisticated trojans, to see the challenge pattern presented to the user, and therefore the solution to that challenge that the user types.

While each challenge pattern is only used once, it is theoretically conceivable that an attacker could record a sufficient number of challenge patterns and their attendant solutions to perform a statistical analysis whereby the key pattern itself could be thus derived.

In the interests of testing PassWindow's vulnerability to such an attack, a cracking algorithm was constructed that attempts to use these principles to perform the said analysis.

The algorithm itself uses a brute-force technique. It begins by enumerating all permutations whereby the solution digits could be placed in a pattern.

For example, a six-digit solution in a 14-column arrangement yields the following possibilities (among others):

2-5-7-2-4-3---

2-5-7-2-4--3--

2-5-7-2-4---3-

2--5--7-2-4--3

Each possibility is evaluated against the known challenge pattern for compatibility or incompatibility on account of the impossibility of representing the digit in question.

Segments may either be **on** if required to construct the solution, **off** if it must be off for the solution, or **unknown** if it falls away from the digits or coincides with a bit in the challenge.

After a separate set of candidates is made for each interception, the algorithm looks for incompatibilities between the candidates. It considers the first candidate of the first set, comparing it in turn to each candidate of the second set. If it is incompatible with every candidate in the second set, the candidate is discarded.

Compatibility checks are continued in this way whereby each candidate in each set is compared to the candidates of every other set. If a candidate is discarded, then every other set needs to be reconsidered.

Eventually, through acquiring and analysing a sufficient number of interceptions, the algorithm is able to deduce the key pattern with reasonable certainty, and in this sense, PassWindow can under idealized conditions fall to this kind of attack.

However, as this attack requires a significant number of interceptions by a dedicated hacker, from 20-30 in the case of small patterns, to hundreds for larger patterns, to many thousands in the case of the high-security animated mode. In the real world, a customer would not normally authenticate using PassWindow frequently or numerous enough to conduct the necessary extended surveillance the attacker requires.

In this way, PassWindow's security resides not so much in the complexity of the algorithm required to solve it, but rather in the systemic difficulty of extracting sufficient information from the target. If PassWindow is employed correctly, the necessary information may very well be impossible for even the most dedicated hacker to acquire. Such a dedicated hacker would likely look for other weaknesses in their victim's personal security to exploit.

Scalable security

PassWindow has two major modes of operation – animated and static challenge patterns:

- **Animated challenge patterns** display a sequence of challenge patterns, each containing a single digit. The user enters a string of sequentially appearing digits to the authentication server. This method is practically immune to brute-force attacks given that the requisite surveillance of sufficient successful authentications would be highly unlikely to occur in a real-world setting.
- **Static challenge patterns** present a single challenge pattern that combines with the customer's key pattern to reveal a multi-digit numeric solution that is submitted to the authentication server. The static pattern method provides a reasonable level of security, and can only be statistically derived following a protracted surveillance period. Static patterns are recommended for use where the higher-security animated method is not possible, as in print-based applications.

In addition, the size and complexity of the PassWindow key and challenge patterns can be varied according to the level of security required. By increasing the number of columns and rows, the security of PassWindow is increased, but at the expense of space on the card that might otherwise be used for other information or decoration; and usability, in that a solution with a greater number of digits must be read.

The larger the key area, the more times a real-time screen-capturing and key-logging hacker would have to successfully intercept the challenge pattern and solution to undertake the statistical analysis required to make a reasonable guess at the challenge pattern. In implementing PassWindow, the level of security required versus the amount of space willing to be devoted to the key pattern is an important consideration.

Other security features

- PassWindow's essentially irreducible simplicity means that any security issues that might have arisen during development were clear and engineered out of the final product.
- The challenge patterns generated by the PassWindow server are computed specifically to thwart attempts at mathematical analysis.
- Physical access to the key pattern printed on the card is reduced by the application of a tint printed over the pattern, rendering it very difficult to photograph or photocopy, while not compromising legibility when held against a back-lit display.

- Challenge patterns are transmitted in a fragmented state making their extraction from a compromised computer's cache more complicated.
- Verification information can be encoded into the challenge pattern, assuring the customer of the transaction's authenticity.
- The only possible route whereby an attacker could plausibly deduce the customer's key pattern – long-term interception and pattern analysis – can be entirely mitigated through authentication management software.

Authentication management

Coupled with authentication management software, PassWindow could be implemented in such a way as to be secure and user-friendly. An authentication manager could perform tasks such as:

- Recording the details of each user's authentication behaviour to increase the intelligibility of the emergent digits for older or visually impaired users
- Monitoring the number and frequency of authentication attempts to reveal problems or aberrant behaviour that might be indicative of hacking attempts
- Issuing a new card once a sufficient number of successful authentications have occurred such that its security is ensured, even in a worst-case scenario

Conclusion

In summary, PassWindow appears to offer a very simple, easy to implement, cost-effective, and secure authentication methodology that if properly implemented, can be resistant to online hacking, including the very latest strategies. No other technology at the time of writing offers a similarly well-rounded package of benefits, certainly not at such a low cost. Due to PassWindow's essentially irreducible simplicity, it is unlikely that any competing technologies may arise in the near future to challenge it on the grounds of cost, administrative burden, or ease of implementation.

While it is conceivable that with the exponential rise of computer processor technology, cryptographic solutions will need to be perpetually made more complex to stay ahead, PassWindow is unlikely to become any less safe in future as its security depends primarily on the user's behaviour.

With these considerations in mind, we feel confident in recommending the PassWindow authentication method as an effective solution for today's online second factor authentication requirements. Of course, each implementation will need to be analysed individually in order to assess their actual security against cryptanalytic and hacking attacks.

About the authors

Sean O'Neil is a world-renowned cryptologist, reverse engineer, and code breaker with 18 years of experience in the industry as an IT security consultant, carrying with him a broad range of security expertise. He is responsible for the security of many products sold by a number of well known, large IT security and antivirus companies protecting corporations, academic, government and financial institutions around the world including the DoD, HSBC, and the Australian Government.

Sean is also responsible for the design of VEST ciphers, EnRUPT, and the method of cryptanalysis known as monomial randomness testing also known as algebraic structure defectoscopy. Both EnRUPT and VEST ciphers have been recognized by the academic community as extremely hardware efficient, flexible and advanced cryptographic designs that differ dramatically only in their software performance and design complexity. EnRUPT, being the simplest block/stream cipher/hash ever made and VEST being the most complex stream cipher/hash ever made, are both used by many cryptology departments of universities as an educational tool for cryptology students.

Peter Lock works at the forefront of the server-security industry as a computer programmer and systems administrator developing cutting-edge server implementations for the corporate market. Heading Cascade Blaze Pty Ltd, Peter also specializes in the provision of statistical and mathematical analysis tools for a number of high-profile clients.

Simon Nettle is a scientific editor and technical writer who has contributed extensively to the fields of medical, scientific, and IT research. At the time of writing he is employed by APNIC, the Internet registry for the Asia Pacific region, and Heurocrypt, an organization that focuses on the research and development of cryptographic communications security techniques and solutions.

Copyright © 2009 by Simon Nettle, Sean O'Neil, Peter Lock